This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

LookasideVLN: Direction-Aware Aerial Vision-and-Language Navigation

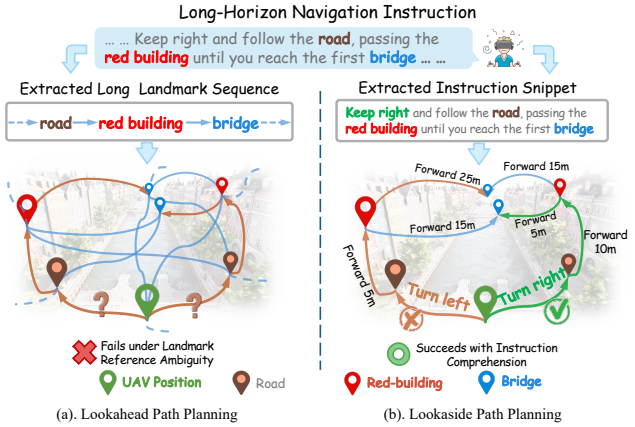
Yuwei Ning[1,2†] Ganlong Zhao[3,4†] Yipeng Qin[5] Si Liu[6] Yang Liu[1] Liang Lin[1,2,7] Guanbin Li[1,2,7*]
[1]Sun Yat-sen University [2]Peng Cheng Laboratory [3]The Chinese University of Hong Kong
[4]Centre for Perceptual and Interactive Intelligence [5]Cardiff University
[6]Beihang University [7]Guangdong Key Laboratory of Big Data Analysis and Processing
ningyw@mail2.sysu.edu.cn, glzhao@cpii.hk, qinyipeng1991@gmail.com, liusi@buaa.edu.cn,
liuy856@mail.sysu.edu.cn, linliang@ieee.org, liguanbin@mail.sysu.edu.cn

Abstract

Aerial Vision-and-Language Navigation (Aerial VLN) enables unmanned aerial vehicles (UAVs) to follow natural language instructions and navigate complex urban environments. While recent advances have achieved progress through large-scale memory graphs and lookahead path planning, they remain limited by shallow instruction understanding and high computational cost. In particular, existing methods rely primarily on landmark descriptions, overlooking directional cues—a key source of spatial context in human navigation. In this work, we propose LookasideVLN, a new paradigm that exploits directional cues in natural language to achieve both more accurate spatial reasoning and greater computational efficiency. LookasideVLN comprises three core components: (1) an Egocentric Lookaside Graph (ELG) that dynamically encodes instruction-relevant landmarks and their directional relationships, (2) a Spatial Landmark Knowledge Base (SLKB) that provides lightweight memory retrieval from prior navigation experiences, and (3) a Lookaside MLLM Navigation Agent that aligns multimodal information from user instructions, visual observations, and landmark-direction information from ELG for path planning. Extensive experiments show that LookasideVLN significantly outperforms the state-of-the-art CityNavAgent, even with a single-level lookahead, demonstrating that leveraging directional cues is a powerful yet efficient strategy for Aerial VLN.

1. Introduction

Aerial Vision-and-Language Navigation (Aerial VLN) [12, 15, 23, 29, 36] has emerged as a novel and challenging task that enables unmanned aerial vehicles (UAVs) to navigate complex aerial environments (*e.g.*, urban scenes) based on natural language instructions, which offers great potential for real-world applications such as urban surveillance, traffic monitoring, disaster response, and aerial delivery.



(a). Lookahead Path Planning (b). Lookaside Path Planning

Figure 1. Comparison of two path planning paradigms. (a) The existing *lookahead* paradigm performs path planning by aligning *long sequences of landmark* descriptions with UAV observations, which is inefficient and fails to fully disambiguate the landmark descriptions. (b) Our *lookaside* paradigm leverages directional cues to enable more efficient and effective path planning.

As a recent addition to the VLN family, Aerial VLN also aims to identify the most appropriate navigation path by aligning UAV's visual observations with natural language instructions. However, large-scale urban environments impose unique challenges absent in small-scale indoor settings. Specifically, early ground-based VLN methods leverage the *uniqueness of landmarks* (a.k.a. objects) to align observations with their textual descriptions [2, 8, 9, 25, 38]. For instance, a UAV can easily follow "go to the kitchen", as most houses contain only one kitchen. In contrast, this property breaks down in urban environments, where landmark descriptions are often *ambiguous* and correspond to multiple instances, e.g., "tree", "wall" or "traffic light". Such ambiguity makes it difficult to infer the correct path by aligning

[†]Equal Contribution.
[*]Corresponding author is Guanbin Li.

visual inputs with individual landmark descriptions. Consequently, early Aerial VLN methods [29, 40] adapted from ground-based VLN exhibit limited performance. Recently, [33, 39] attempt to address this by assuming the *uniqueness of landmark sequences* and constructing a global scene graph for sequence-level alignment. While mitigating the issue to some extent, this approach is inefficient due to the heavy computational and memory costs of global graph maintenance in large-scale environments.

In this work, we introduce LookasideVLN, a new *lookaside* paradigm that addresses the above limitations by leveraging **directional cues** naturally embedded in user instructions for path planning, in contrast to prior *lookahead*-based approaches that depend on long landmark-sequence alignment. As illustrated in Fig. 1, our key insight is that *such directional cues encode rich spatial information, enabling more effective and efficient disambiguation of landmark descriptions in natural language instructions*. To realize this idea, we design a UAV navigation system that exploits directional cues through a modular architecture comprising three main components: (1) An **Egocentric Lookaside Graph (ELG)**. For the "egocentric" design, our rationale stems from the fact that directional cues in natural language instructions (e.g., "turn left," "fly straight ahead," "go past the building on your right") are inherently egocentric, grounded in the navigator's viewpoint rather than a global, map-based frame. Accordingly, we exploit these cues by dynamically constructing an egocentric lookaside graph (ELG) for each instruction. The ELG adopts a hierarchical structure mirroring the sequential landmark descriptions, where each level corresponds to an extracted landmark description and nodes within the same level represent candidate landmarks—visually similar locations retrieved from the UAV's memory. Owing to this egocentric formulation, the ELG is significantly more efficient than a global scene graph, as it models only instruction-relevant landmarks. For the "lookaside" design, we further model directional relationships along inter-hierarchy edges relative to the UAV's egocentric orientation, enabling effective spatial reasoning and path planning within a compact graph structure. (2) A **Spatial Landmark Knowledge Base (SLKB)**, which serves as the UAV's memory for constructing our ELG. Specifically, it stores descriptions and locations of previously visited landmarks extracted from past navigation experiences. Our SLKB is both lightweight, as it omits explicit inter-landmark relationships, and efficient, as it enables fast retrieval via a hierarchical (description, location) structure. Moreover, inspired by *Liebig's Law of the Minimum*, we note that the linguistic modality of user instructions inherently constrains the exploitable information for language-vision alignment, rendering detailed visual features used in previous methods [33, 39] unnecessary. (3) A **Lookaside MLLM Navigation Agent**, which employs a multimodal large language model (MLLM) to align the UAV's current observation, user instruction, and the *directional landmark information* provided by the ELG for navigation. This enables comprehensive multimodal alignment, leveraging both semantic and lookaside directional cues in the user instruction to guide navigation more effectively. Extensive experiments demonstrate the effectiveness of our lookaside paradigm. Specifically, our LookasideVLN, with only a single level of lookahead planning, surpasses the state-of-the-art CityNavAgent [39], despite the latter performing full landmark-sequence lookahead over the entire urban scene.

In summary, our contributions include:

- We introduce a novel paradigm, namely LookasideVLN, which leverages directional cues in natural language instructions to enhance spatial reasoning and mitigate landmark description ambiguity effectively.
- To implement LookasideVLN, we develop a UAV navigation system consisting of three novel components: an Egocentric Lookaside Graph (ELG), a Spatial Landmark Knowledge Base (SLKB), and a Lookaside MLLM Navigation Agent.
- Extensive experimental results show that our LookasideVLN achieves state-of-the-art performance, outperforming CityNavAgent [39] even with a single-level lookahead planning, highlighting the surprising effectiveness of lookaside directional cues.

2. Related Work

Vision-and-Language Navigation. Vision-and-Language Navigation (VLN) is a core task in embodied AI, first introduced by the Room-to-Room (R2R) benchmark [2], where agents navigate discrete indoor environments. R2R-CE [21] extends VLN to continuous settings, bridging the gap between simulation and real-world navigation. Early approaches [8, 9, 13, 16, 18, 19, 25, 28] rely on end-to-end trained models that align language instructions with visual observations to predict actions. However, these approaches often suffer from error accumulation during sequential action prediction, causing the agent's trajectory to gradually drift away from the intended route.

Liu et al. [29] introduce Aerial VLN, a more challenging benchmark that requires agents to navigate in continuous aerial environments. They adapt the cross-modal attention mechanism from ground-based VLN and employ the look-ahead training paradigm [32], in place of the previously used student-forcing strategy [2]. Subsequent efforts have been made toward Aerial VLN. For instance, Zhao et al. [40] introduce a grid-based view selection mechanism and a top-down map representation to facilitate effective aerial navigation. Nonetheless, these methods inherit the limitation of error accumulation from ground-based VLN, limiting their effectiveness in long-horizon naviga-

tion scenarios. Other works explore zero-shot LLM-based approaches for Aerial VLN by decomposing long-horizon navigation tasks into sequential subgoals, and developing specialized scene modeling techniques. STMR [14] introduces an LLM-readable semantic top-down map by projecting semantically segmented point clouds into a textual matrix, serving as the agent's scene perception. However, this top-down representation compresses vertical (height) information, which is crucial for effective aerial navigation. CityNavAgent [39], inspired by LM-Nav [33], adopts lookahead path planning by incorporating a graph search algorithm over a city-scale scene graph, allowing for more holistic reasoning. Nevertheless, it primarily relies on the semantic similarity between landmark descriptions and visual observations, while ignoring directional cues and lacking a thorough understanding of the instructions. In this work, we address this with a novel *lookaside* paradigm.

LLM for Embodied Planning. Recently, Large Language Models (LLMs) [1, 17, 31, 34, 35, 41] have demonstrated remarkable capabilities in language understanding and reasoning, attracting widespread interest in their application to robotics. In particular, LLMs have been employed for embodied task planning and motion planning. Some studies leverage the zero-shot or few-shot prediction capabilities of LLMs to generate static plans for embodied agents [6, 11, 20], while others enable agents to adapt their plans or actions in response to environmental feedback [10, 42].

More recent work has explored memory-augmented approaches for embodied planning. EmbodiedRAG [5] is the first to integrate Retrieval-Augmented Generation (RAG) [24] into embodied planning, leveraging subgraphs retrieved from scene graphs to reduce run-time computational overhead. RAG-Driver [37] enhances explainable autonomous driving by retrieving driving experiences similar to the current scenario from a memory database. In the context of indoor VLN, KERM [25] and Bao et al. [4] adopt RAG by retrieving textual descriptions relevant to the agent's observations from an external image caption dataset [22]. However, relying on external datasets introduces domain gaps, which can hinder robust navigation performance. In this work, we construct a Spatial Landmark Knowledge Base comprising landmark descriptions and positions, serving as a spatial memory module for the agent.

3. Method

In this section, we introduce **LookasideVLN**, a systematic navigation framework which leverages *directional cues* naturally embedded in natural language instructions for path planning, implementing a novel *lookaside* paradigm. As illustrated in Fig. 2, LookasideVLN first queries the Spatial Landmark Knowledge Base (SLKB) using landmarks extracted from navigation instructions to retrieve locations for each candidate landmark (Sec. 3.1). It then constructs the Egocentric Lookaside Graph (ELG) to model both landmarks and their directional relationships (Sec. 3.2.1). Next, directional paths on the ELG are converted into instruction-like descriptions, enabling the agent to evaluate their alignment with user instructions (Sec. 3.2.2). Finally, to support effective path planning, we introduce a Lookaside MLLM Navigation Agent, which makes path decisions in a robust and explainable manner by jointly reasoning over the scene understanding and navigation instructions (Sec. 3.3).

3.1. Spatial Landmark Knowledge Base

Our **Spatial Landmark Knowledge Base (SLKB)** is an efficient, lightweight, and scalable memory module built from both scene observations and historical navigation records, enables the agent to recall previously observed landmarks and progressively accumulate scene knowledge over time.

3.1.1. Lightweight and Scalable Design

To reduce redundancy and achieve efficient retrieval, we design our SLKB $\mathcal{K}$ hierarchically as:

$$\mathcal{K} = \left\{ l_i^{kb} : \{p_{i,0}^{kb}, p_{i,1}^{kb}, \dots, p_{i,j}^{kb}\} \right\}, \tag{1}$$

where l_i^{kb} is the i-th landmark description, $p_{i,j}^{kb}$ denotes the j-th candidate location for l_i^{kb}. Since our SLKB is a dynamically updated memory, we show below how a new entry is inserted or merged (Fig. 3). Historical navigation records are incorporated in the same way.

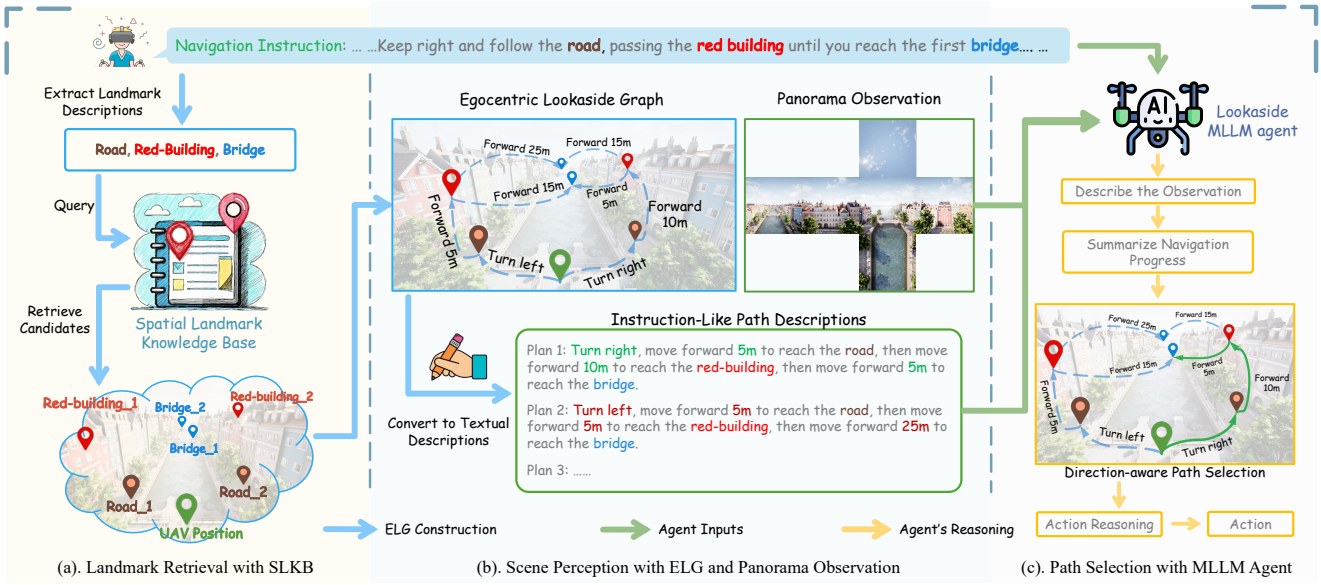
Extraction of Newly-observed l^{kb} and p^{kb}. Given an RGB observation I_{rgb} from the scene, we use a MLLM-driven *Landmark Recognizer* $\mathrm{LR}(\cdot)$ to generate descriptions l_i^{kb} for each landmark in I_{rgb}. Next, we use a *Landmark Detector* $\mathrm{LD}(\cdot)$, empowered by GroundingDINO [30], to predict bounding boxes $\mathcal{B} = \{b_1, b_2, \dots\}$ for l_i^{kb}, *i.e.*,

$$\mathcal{B} = \mathrm{LD}(l_i^{kb}). \tag{2}$$

We then filter redundant, overlapping bounding boxes using Non-Maximum Suppression (NMS) to obtain the final bounding box b and compute its 3D location from the corresponding depth observation I_{depth} of I_{rgb}. Given the center pixel coordinate (u, v) of bounding box b and the mean depth value $\bar{d}$ within the box (computed after removing outliers beyond 2σ from the depth distribution), the landmark location $p_i^{kb} \in \mathbb{R}^3$ in world coordinates is calculated as:

$$\begin{aligned} p_i^{kb} &= \frac{\bar{d}_i}{\|K^{-1} p_i^{pixel}\|_2} \cdot R K^{-1} p_i^{pixel} + T, \\ p_i^{pixel} &= [u, v, 1]^\top \end{aligned} \tag{3}$$

where $R \in SO(3)$ and $T \in \mathbb{R}^3$ are the camera's rotation matrix and translation vector, while $K \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix. Note that we omit landmark descriptions from user instructions, as they are already covered by



(a). Landmark Retrieval with SLKB (b). Scene Perception with ELG and Panorama Observation (c). Path Selection with MLLM Agent

Figure 2. Overall framework of LookasideVLN. The agent queries the Spatial Landmark Knowledge Base to construct an Egocentric Lookaside Graph, which is then converted into instruction-like, direction-aware path descriptions. A Lookaside MLLM agent selects the final path based on the alignment among the directional path, instruction snippet, and current observation.

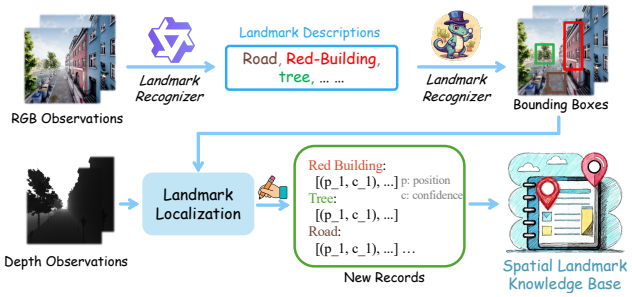


Figure 3. Spatial Landmark Knowledge Base Construction.

those extracted from RGB observations, which also provide the corresponding landmark positions. Further implementation details are provided in the supplementary materials.

Insertion or Merging of (l^{kb}, p^{kb}). Following the standard database operations, given a new landmark–position pair (l^{kb}, p^{kb}), our SLKB $\mathcal{K}$ determines whether l^{kb} corresponds to an existing entry. If a similar landmark already exists, the new p^{kb} is merged to update the existing entry of l^{kb}. Otherwise, a new record is inserted into the SLKB. This strategy ensures that the knowledge base remains compact and up to date while avoiding redundant entries and preserving spatial consistency across navigation experiences.

Remark on Compact Landmark Description l^{kb}. Note that, inspired by *Liebig's Law of the Minimum* (a.k.a., the barrel theory), we argue that the linguistic modality of user instructions inherently limits the exploitable information for language–vision alignment, making the detailed visual features used in prior methods [33, 39] unnecessary. Accordingly, we represent each landmark l_i^{kb} textually, reducing memory and computational overhead.

3.1.2. Efficient Data Retrieval

As shown in Fig. 2, given our hierarchical SLKB $\mathcal{K}$, for each landmark description extracted from the user instruction $\{l_i^{instr}\}_{i=1}^{N}$, we compute its word embedding [26] and compare it with the embeddings of all landmark descriptions l_j^{kb} in $\mathcal{K}$ using cosine similarity $\mathrm{sim}(\cdot)$, retrieving the one with maximum similarity:

$$l_i^{ret} = \mathrm{argmax}_{l_j^{kb} \in \mathcal{K}} \, \mathrm{sim}\big(\mathrm{emb}(l_i^{instr}), \mathrm{emb}(l_j^{kb})\big), \quad (4)$$

where $\mathrm{emb}(\cdot)$ is the word embedding model. Then, we can efficiently retrieve all positions of each landmark satisfying the description l_i^{ret} as:

$$\mathcal{L}^{ret} = \big\{ l_i^{ret} : \{p_{i,0}^{ret}, p_{i,1}^{ret}, \dots\} \big\}_{i=1}^{N}. \quad (5)$$

3.2. Egocentric Lookaside Graph

Navigation instructions typically adopt an egocentric perspective — such as *"turn left and go to the intersection, then turn right and go to the bridge"* — where directional cues are essential for disambiguating between multiple candidate landmarks of the same category (e.g., multiple bridges in the environment). In such instructions, directional cues like *"turn right"* are defined relative to **the agent's orientation upon reaching the preceding key landmark** (i.e. *the intersection* in this case). We refer to these as **egocentric**

lookaside directional relationships, since they depend on the agent's future orientation after passing key landmarks in the instruction. To effectively leverage directional cues in navigation instructions, we propose the **Egocentric Lookaside Graph (ELG)**, where nodes represent candidate locations of unvisited landmarks mentioned in the instruction, and edges contain egocentric lookaside directional relationships between them. Unlike previous works [33, 39] that rely solely on proximity in large-scale scene graphs, our ELG dynamically captures instruction-relevant landmarks and directions to construct small egocentric graphs, where paths can be seamlessly translated into instruction-like descriptions, facilitating semantic scene understanding and effective path planning.

3.2.1. Egocentric Lookaside Graph (ELG) Construction

Starting from the UAV's current position, we construct the ELG $\mathcal{G}(\mathcal{N}, \mathcal{E})$ by selecting the next N_{ahead} *unvisited* landmarks from $\mathcal{L}^{ret}$ (Eq. 5) to form:

$$\mathcal{L}^{unvis} = \left\{ l_i^{unvis} : \{p_{i,0}^{unvis}, p_{i,1}^{unvis}, \dots\} \right\}_{i=1}^{N_{ahead}}, \quad (6)$$

where l_i^{unvis} is the i-th unvisited landmark and $p_{i,j}^{unvis}$ is the j-th candidate location of l_i^{unvis}. We then construct the ELG hierarchically, where each layer corresponds to a landmark description l_i^{unvis} in sequence, and each node within a layer represents a unique location $p_{i,j}^{unvis}$. The collection of all nodes forms $\mathcal{N}$. For each pair of successive unvisited landmark layers (l_{i-1}^{unvis} and l_i^{unvis}) in the sequence, we create edges between every candidate location $p_{i-1,j}^{unvis}$ of the preceding landmark and every candidate location $p_{i,k}^{unvis}$ of the subsequent landmark. These edges are then assigned their corresponding egocentric lookaside directional relationships as follows.

Egocentric Direction Estimation from Location. Considering the candidate locations of three consecutive landmarks $(p_{i-1,j}^{unvis}, p_{i,k}^{unvis}, p_{i+1,m}^{unvis})$, we first estimate the agent's orientation by computing the direction from $p_{i-1,j}^{unvis}$ to $p_{i,k}^{unvis}$. This estimated orientation is then used to determine the egocentric lookaside directional relationship between $p_{i,k}^{unvis}$ and $p_{i+1,m}^{unvis}$. Specifically, the agent's orientation upon reaching $p_{i,k}^{unvis}$ is estimated as the unit vector:

$$\mathbf{p}_{i-1,j}^{i,k} = \frac{p_{i,k}^{unvis} - p_{i-1,j}^{unvis}}{\|p_{i,k}^{unvis} - p_{i-1,j}^{unvis}\|_2}, \quad (7)$$

where $\mathbf{p}_{i-1,j}^{i,k} \in \mathbb{R}^3$ represents the agent's viewing direction after transitioning from $p_{i-1,j}^{unvis}$ to $p_{i,k}^{unvis}$. Then we compute the agent's horizontal deflection angle $\theta_{i,k}^{i+1,m}$, elevation gain $e_{i,k}^{i+1,m}$, and relative horizontal distance $d_{i,k}^{i+1,m}$ from candidate location $p_{i,k}^{unvis}$ to $p_{i+1,m}^{unvis}$ as follows:

$$\begin{aligned} \theta_{i,k}^{i+1,m} &= \text{hangle}(\mathbf{p}_{i-1,j}^{i,k}, \mathbf{p}_{i,k}^{i+1,m}) \\ e_{i,k}^{i+1,m} &= |(p_{i+1,m}^{unvis} - p_{i,k}^{unvis})_z| \\ d_{i,k}^{i+1,m} &= \|(p_{i+1,m}^{unvis} - p_{i,k}^{unvis})_{xy}\|_2, \end{aligned} \quad (8)$$

where $\text{hangle}(\mathbf{v}_1, \mathbf{v}_2)$ [*] computes the horizontal deflection angle (in the xy-plane) from $\mathbf{v}_1$ to $\mathbf{v}_2$, $(\cdot)_z$ extracts the vertical component and $(\cdot)_{xy}$ denotes the horizontal projection of a vector. We utilize the variables defined in Eq. 8 to represent the egocentric lookaside directional relationships between candidate landmarks, enabling the ELG to be seamlessly translated into instruction-like path descriptions for downstream MLLM-based semantic-level path planning.

3.2.2. Generating Path Descriptions from ELG

To enable the agent to better align instruction understanding with scene comprehension, we convert each possible path in the ELG into an instruction-like description. These descriptions, combined with the navigation instruction, allow the agent to reason and perform path planning in a more intuitive and language-aligned manner. Given $\theta_{i,k}^{i+1,m}$, $e_{i,k}^{i+1,m}$ and $d_{i,k}^{i+1,m}$ defined in Eq. 8, we describe the motion from $p_{i,k}^{unvis}$ to $p_{i+1,m}^{unvis}$ of the unvisited landmark l_{i+1}^{unvis} as:

> "Turn left/right $|\theta_{i,k}^{i+1,m}|$ degrees, move forward $d_{i,k}^{i+1,m}$ meters and ascend/descend $e_{i,k}^{i+1,m}$ meters to reach l_{i+1}^{unvis}."

This detailed form is used for the first unvisited landmark. For later steps, we use a coarser description:

> "Turn left/right $|\theta_{i,k}^{i+1,m}|$ degrees and move toward the l_{i+1}^{unvis}."

By traversing all possible paths in the ELG, we generate a set of full path descriptions $\mathcal{P}$, where each entry describes a candidate path. An example from $\mathcal{P}$ is as follows:

> "Turn left 30 degrees, move forward 10 meters and descend 4 meters to reach the intersection, then turn right 45 degrees and proceed toward the bridge, ..."

All such descriptions are compiled into a prompt to support semantic-level path planning. We also apply a distance-based pruning strategy to improve reasoning efficiency. The details are provided in the supplementary materials.

3.3. Lookaside MLLM Navigation Agent

We introduce a Lookaside MLLM Navigation Agent that achieves robust and interpretable path planning. As illustrated in Fig. 2, our MLLM agent processes the navigation instruction $\mathcal{I}$, the path descriptions $\mathcal{P}$ derived from

[*] $\text{hangle}(\mathbf{v}_1, \mathbf{v}_2) = \text{atan2}((\mathbf{v}_1 \times \mathbf{v}_2)_z, \mathbf{v}_1 \cdot \mathbf{v}_2 - v_{1x} v_{2x})$

the ELG, and the current panoramic observation $O_t = \{o_{t,i}\}_{i=1}^{6}$ — comprising front, left, right, back, top, and bottom view images — to sequentially: 1) generate an Observation Description, 2) summarize the Navigation Progress, 3) perform Direction-aware Path Planning, and 4) conduct Action Reasoning to determine the next action a_t.

3.3.1. Observation Description and Navigation Progress

At the start of the chain-of-thought reasoning, we prompt the agent to generate an Observation Description, which helps it understand its surroundings by producing a textual summary of the input observation images. In addition, following the approach of prior works [14, 43], we also require the agent to summarize its navigation progress, enabling it to assess and track the current stage of the task.

3.3.2. Direction-aware Path Planning

To align the agent's navigation decisions with natural language instructions, we propose a Direction-aware Path Planning strategy. The objective is to select an appropriate direction-aware path description from the candidate set $\mathcal{P}$ on the ELG, guided by the navigation instruction $\mathcal{I}$. Specifically, the agent is instructed to: 1) identify the unvisited landmarks within the candidate paths $\mathcal{P}$; 2) extract the corresponding snippets from $\mathcal{I}$ that describe how to reach these landmarks; and 3) select the most suitable path from $\mathcal{P}$ based on the grounded instruction segments.

3.3.3. Action Reasoning

After selecting a path from the ELG, the agent performs an Action Reasoning process to determine the next action a_t. This process is guided by the next unvisited landmark on the selected path and further refined using the instruction $\mathcal{I}$ and current visual observation O_t. Detailed prompts are provided in the supplementary materials.

4. Experiment

Dataset We evaluate our LookasideVLN on the challenging AerialVLN benchmark [29], which comprises 8,446 flight trajectories collected from experienced UAV pilots across 25 diverse city-scale environments in Unreal Engine 4. AerialVLN emphasizes long-horizon navigation, with an average path length of 661.8 meters. Each episode consists of an aerial trajectory with a corresponding natural language navigation instruction. The validation set is partitioned into "Seen" and "Unseen" splits according to the presence of each scene in the training set. AerialVLN-S is a small-scale variant of AerialVLN, consisting of 17 compact scenes.

Metrics Following the evaluation metrics of AerialVLN, we assess LookasideVLN's performance using four standard metrics: **Success Rate (SR)**, which counts a navigation as successful if the agent stops within 20 meters of the destination; **Oracle Success Rate (OSR)**, which considers a navigation as oracle success if any point on the trajectory is within 20 meters of the destination; **Navigation Error (NE)**, defined as the distance between the agent's stopping point and the destination; and **Success Rate weighted by Normalized Dynamic Time Warping (SDTW)**, which accounts for both task success and the similarity between predicted and ground-truth trajectories.

Baselines We compare LookasideVLN with a range of baselines across four categories. *Statistical methods*, including **Random** and **Action Sampling** [29], serve to illustrate the size of the solution space and the similarity in action distributions between the training and evaluation splits. *Learning-based methods* cover traditional end-to-end action prediction models such as **Seq2Seq** [2], **CMA** [21], **LAG** [29], and the method of **Zhao et al.** [40] (denoted as zhao'25 in the tables). To assess generalization from indoor domains, we also include *zero-shot indoor VLN methods* such as **NavGPT** [43] and **MapGPT** [7]. Finally, we compare with *zero-shot Aerial VLN methods*, including the state-of-the-art **STMR** [14] and **CityNavAgent** [39]. Note that, due to the challenging nature of the AerialVLN dataset, existing zero-shot baselines cannot be directly applied. Thus, we primarily compare against learning-based methods on AerialVLN and include zero-shot baselines in the evaluation on AerialVLN-S.

Implementation Details We implement LookasideVLN based on Qwen2.5-VL-72B [3], which is accessed through its online API. To construct the Spatial Landmark Knowledge Base (Sec. 3.1), we randomly sample 50 trajectories from the training set for each seen scene. For the unseen split, we pre-render images across the environment to serve as scene observations. The Landmark Recognizer is built upon Qwen-VL-Max [3] and accessed via online APIs. Unless otherwise specified, we set the lookahead parameter N_{ahead} in the ELG to 2. For validation, the agent is initialized at the starting point of each episode. At each step, it selects one of six discrete actions: *Move Forward*, *Turn Left*, *Turn Right*, *Ascend*, *Descend*, or *Stop*, along with the number of times the selected action should be executed. Each action moves or rotates the agent as follows: *Turn Left/Right* 15 degrees, *Ascend/Descend* 2 meters vertically, and *Move Forward* 5 meters. Navigation ends when the agent chooses the *Stop* action or reaches the maximum number of steps.

4.1. Experimental Results

4.1.1. Quantitative Evaluation

Table 1 presents the comparative results on the AerialVLN benchmark. Our *training-free* LookasideVLN surpasses traditional *learning-based* methods such as Seq2Seq and CMA by a clear margin and achieves performance comparable to Zhao'25 on the "Seen" split. Notably, unlike Zhao'25 and other learning-based approaches, our method exhibits strong generalization capability, achieving higher performance on the "Unseen" split. This underscores the ro-

Category	Method	Validataion Seen				Validation Unseen			
		SR↑	OSR↑	SDTW↑	NE↓	SR↑	OSR↑	SDTW↑	NE↓
Statistical	Random	0.0	0.0	0.0	300.8	0.0	0.0	0.0	351.0
	Action Sampling	0.1	2.1	0.1	383.1	0.2	2.1	0.1	434.9
Learning-Based	Seq2Seq	2.9	10.2	1.0	480.4	1.1	5.6	0.3	551.8
	CMA	2.3	6.5	0.8	293.5	1.6	4.4	0.5	360.7
	Zhao'25	**7.5**	12.6	**3.3**	**271.1**	3.2	8.1	**1.3**	**333.5**
Zero-Shot	LookasideVLN (Ours)	5.7	**26.1**	1.2	278.6	**6.4**	**21.3**	1.2	487.0

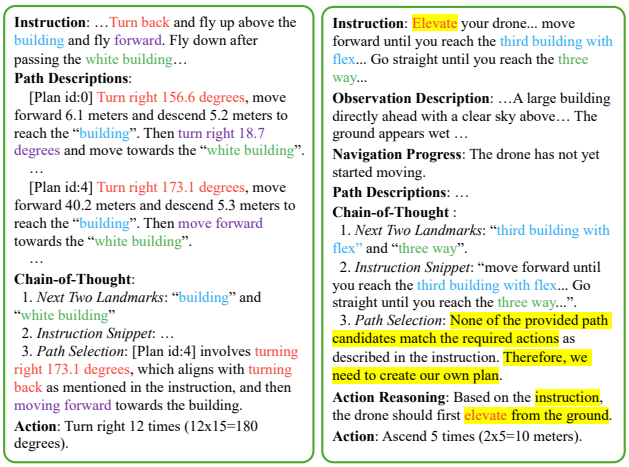
Table 1. Overall performance comparison on the AerialVLN benchmark. Bold and underline indicate the first and second best results.

Category		Method	LLM	Validataion Seen				Validation Unseen			
				SR↑	OSR↑	SDTW↑	NE↓	SR↑	OSR↑	SDTW↑	NE↓
Statistical		Random	-	0.0	0.0	0.0	109.6	0.0	0.0	0.0	149.7
		Action Sampling	-	0.9	5.7	0.3	213.8	0.2	1.1	0.1	237.6
Learning-Based		Seq2Seq	-	4.8	19.8	1.6	146.0	2.3	11.7	0.7	218.9
		CMA	-	3.0	23.2	0.6	121.0	3.2	16.0	1.1	172.1
		LAG	-	7.2	15.7	2.4	90.2	5.1	10.5	1.4	127.9
Zero-Shot	Indoor	NavGPT	GPT-4	0.0	0.0	0.0	163.5	0.0	0.0	0.0	82.1
		MapGPT	GPT-4V	2.1	4.7	0.8	124.9	0.0	0.0	0.0	107.0
	Aerial	STMR	GPT-4o	12.6	**31.6**	-	96.3	10.8	23.0	-	119.5
		CityNavAgent	GPT-4V	13.9	30.2	5.1	80.8	11.7	35.2	**5.0**	**60.2**
		LookasideVLN (Ours)	Qwen2.5-VL-72B	**14.7**	31.2	**5.4**	**77.1**	**12.6**	**36.0**	3.6	100.9

Table 2. Overall performance comparison on the AerialVLN-S benchmark. Bold and underline indicate the first and second best results.

bustness and generalizability of our zero-shot formulation.

Table 2 shows the comparison results on the AerialVLN-S benchmark. LookasideVLN, based on the relatively small Qwen2.5-VL-72B model, outperforms prior methods in terms of success rate (SR) on both "Seen" and "Unseen" splits. Several observations can be drawn from the results: 1) Statistical methods like Random and Action Sampling perform worst, highlighting the large solution space and action distribution mismatch between training and validation. 2) Learning-based methods perform poorly in this challenging setting. Most of them suffer a substantial performance drop in the "Unseen" set, suggesting limited generalizability, which is likely due to insufficient training data. 3) Zero-shot indoor VLN methods perform poorly in the Aerial VLN task, emphasizing the greater complexity and distinct challenges of outdoor aerial environments compared to indoor scenes. 4) Despite using a smaller MLLM, LookasideVLN surpasses previous state-of-the-art Aerial VLN methods on most key metrics. This demonstrates the effectiveness of the Egocentric Lookaside Graph and the path planning strategy based on the Lookaside MLLM Agent.



(a) The agent selects a candidate path and determines an action based on the chosen path.

(b) The agent refines its selected action by incorporating both the instruction and observation.

Figure 4. Examples of the agent's action selection process.

4.1.2. Qualitative Evaluation

Fig. 4 presents two examples of the MLLM Agent planning results. In Fig. 4 (a), the agent selects the candidate path

ELG	MLLM Agent	SR↑	SDTW↑	NE↓
×	×	2.4	1.0	405.5
✓	×	13.8	4.6	81.6
✓	✓	14.7	5.4	77.1

Table 3. Ablation study on the Egocentric Lookaside Graph (ELG) and the Lookaside MLLM Navigation Agent.

N_{ahead}	SR↑	SDTW↑	NE↓
1	9.9	3.4	84.9
2	14.7	5.4	77.1
3	11.7	3.9	83.9

Table 4. Ablation study on lookahead horizon N_{ahead}.

with a relatively large turning angle, aligning with the instruction "turn back." In Fig. 4 (b), the agent attempts to find a matching path, but none of the candidates align with the instruction. Recognizing the cue "elevate" and observing that it is on the ground, the agent chooses the *Ascend* action instead. These examples demonstrate that our LookasideVLN can determine actions either through path selection or by aligning with the instruction and current observations.

4.2. Ablation Study

To comprehensively analyze the performance of LookasideVLN, we conduct ablation studies on its key components, critical hyperparameters, and the choice of MLLM used in our method. All ablation experiments are performed on the validation "Seen" split of the AerialVLN-S benchmark.

4.2.1. Effect of Egocentric Lookaside Planning

Table 3 presents an ablation study that evaluates the effectiveness of our proposed modules, including the Egocentric Lookaside Graph (ELG) and the Lookaside MLLM Navigation Agent. We observe that removing the ELG and replacing the agent's reasoning with direct action prediction results in the lowest performance, indicating the significance of both modules. Adding ELG alone improves performance, suggesting that the incorporation of lookaside directional cues significantly benefits long-horizon planning. When both ELG and the agent are included, LookasideVLN achieves the best overall performance, suggesting that incorporating ELG and the agent's reasoning for path planning better support decision-making in Aerial VLN tasks.

4.2.2. Impact of Lookahead Horizon N_{ahead}

We further investigate the effect of the lookahead horizon N_{ahead}, which determines the number of future subgoals are considered when constructing the ELG. As shown in Table 4, setting $N_{ahead} = 2$ yields the best overall performance. When $N_{ahead} = 1$, the agent does not incorporate future subgoal information and behaves similarly to a step-by-step planning method with no explicit lookahead, limiting its ability to make effective path planning. In contrast, when $N_{ahead} = 3$, the resulting path descriptions become longer and more complex, placing a greater burden on the model's reasoning capability. This suggests that a moderate lookahead horizon strikes a balance between enhanced spatial modeling and reduced reasoning complexity. Nevertheless, it is worth noting that our LookasideVLN outperforms the state-of-the-art CityNavAgent [39] even with a small $N_{ahead} = 2$, highlighting the surprising effectiveness of the proposed lookaside directional cues.

MLLM	SR↑	SDTW↑	NE↓
LLaVA-7B	N/A	N/A	N/A
Qwen2.5-VL-7B	9.0	1.7	306.6
Qwen2.5-VL-32B	14.1	4.9	94.3
Qwen2.5-VL-72B (ours)	14.7	5.4	77.1

Table 5. Ablation study on different MLLMs.

4.2.3. Effect of Different MLLMs

We evaluate the impact of different MLLMs on overall performance, as shown in Table 5. LLaVA-7B [27] fails to follow prompts and consistently outputs only observation descriptions, likely due to its limited instruction-following capabilities in long-horizon navigation settings. Qwen2.5-VL-7B, though lightweight, exhibits limited ability to handle the complex long-horizon reasoning required for Aerial VLN. In contrast, both the 32B and 72B versions of Qwen2.5-VL achieve significantly stronger performance. These results suggest that more capable MLLMs provide better support for effective path reasoning.

5. Conclusion

In this work, we introduced LookasideVLN, a new Aerial VLN paradigm that leverages directional cues in natural language to enhance spatial reasoning and efficiency. Unlike prior lookahead-based approaches that rely heavily on costly landmark sequence alignment, LookasideVLN integrates an Egocentric Lookaside Graph, a Spatial Landmark Knowledge Base, and a Lookaside MLLM Navigation Agent to achieve lightweight yet context-aware path planning. Through extensive experiments, we demonstrated that LookasideVLN substantially outperforms the state-of-the-art CityNavAgent, despite using only single-level lookahead planning. These results highlight the value of incorporating directional semantics as a complementary spatial cue in Vision-and-Language Navigation.

Acknowledgments

This work is supported in part by the National Key R&D Program of China (NO. 2024YFB3908503 and 2024YFB3908500), and in part by the National Natural Science Foundation of China (NO. 62322608).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018. 1, 2, 6

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 6

[4] Xiaoan Bao, Zhiqiang Lv, and Biao Wu. Enhancing large language models with rag for visual language navigation in continuous environments. *Electronics*, 14(5):909, 2025. 3

[5] Meghan Booker, Grayson Byrd, Bethany Kemp, Aurora Schmidt, and Corban Rivera. Embodiedrag: Dynamic 3d scene graph retrieval for efficient and scalable robot task planning. *arXiv preprint arXiv:2410.23968*, 2024. 3

[6] Yue Cao and CS Lee. Robot behavior-tree-based task generation with large language models. *arXiv preprint arXiv:2302.12927*, 2023. 3

[7] Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee K. Wong. Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. 6

[8] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Advances in neural information processing systems*, 34:5834–5847, 2021. 1, 2

[9] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16537–16547, 2022. 1, 2

[10] Yongchao Chen, Jacob Arkin, Yang Zhang, Nicholas Roy, and Chuchu Fan. Scalable multi-robot collaboration with large language models: Centralized or decentralized systems? In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4311–4317. IEEE, 2024. 3

[11] Norman Di Palo, Arunkumar Byravan, Leonard Hasenclever, Markus Wulfmeier, Nicolas Heess, and Martin Riedmiller. Towards a unified agent with foundation models. *arXiv preprint arXiv:2307.09668*, 2023. 3

[12] Yue Fan, Winson Chen, Tongzhou Jiang, Chun Zhou, Yi Zhang, and Xin Eric Wang. Aerial vision-and-dialog navigation. *arXiv preprint arXiv:2205.12219*, 2022. 1

[13] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. *Advances in neural information processing systems*, 31, 2018. 2

[14] Yunpeng Gao, Zhigang Wang, Linglin Jing, Dong Wang, Xuelong Li, and Bin Zhao. Aerial vision-and-language navigation via semantic-topo-metric representation guided llm reasoning. *arXiv preprint arXiv:2410.08500*, 2024. 3, 6

[15] Yunpeng Gao, Chenhui Li, Zhongrui You, Junli Liu, Zhen Li, Pengan Chen, Qizhi Chen, Zhonghan Tang, Liansheng Wang, Penghui Yang, et al. Openfly: A versatile toolchain and large-scale benchmark for aerial vision-language navigation. *arXiv e-prints*, pages arXiv–2502, 2025. 1

[16] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1634–1643, 2021. 2

[17] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 3

[18] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1643–1653, 2021. 2

[19] Aishwarya Kamath, Peter Anderson, Su Wang, Jing Yu Koh, Alexander Ku, Austin Waters, Yinfei Yang, Jason Baldridge, and Zarana Parekh. A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10813–10823, 2023. 2

[20] Shyam Sundar Kannan, Vishnunandan LN Venkatesh, and Byung-Cheol Min. Smart-llm: Smart multi-agent robot task planning using large language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12140–12147. IEEE, 2024. 3

[21] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pages 104–120. Springer, 2020. 2, 6

[22] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 3

[23] Jungdae Lee, Taiki Miyanishi, Shuhei Kurita, Koya Sakamoto, Daichi Azuma, Yutaka Matsuo, and Nakamasa Inoue. Citynav: Language-goal aerial navigation dataset with geographic information. *arXiv preprint arXiv:2406.14240*, 2024. 1

[24] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020. 3

[25] Xiangyang Li, Zihan Wang, Jiahao Yang, Yaowei Wang, and Shuqiang Jiang. Kerm: Knowledge enhanced reasoning for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2583–2592, 2023. 1, 2, 3

[26] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023. 4

[27] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024. 8

[28] Rui Liu, Wenguan Wang, and Yi Yang. Volumetric environment representation for vision-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16317–16328, 2024. 2

[29] Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. Aerialvln: Vision-and-language navigation for uavs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15384–15394, 2023. 1, 2, 6

[30] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024. 3

[31] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. 3

[32] Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel X Chang. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. *arXiv preprint arXiv:2109.15207*, 2021. 2

[33] Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pages 492–504. PMLR, 2023. 2, 3, 4, 5

[34] Xinshuai Song, Weixing Chen, Yang Liu, Weikai Chen, Guanbin Li, and Liang Lin. Towards long-horizon vision-language navigation: Platform, benchmark and method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12078–12088, 2025. 3

[35] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3

[36] Xiangyu Wang, Donglin Yang, Ziqin Wang, Hohin Kwan, Jinyu Chen, Wenjun Wu, Hongsheng Li, Yue Liao, and Si Liu. Towards realistic uav vision-language navigation: Platform, benchmark, and methodology. *arXiv preprint arXiv:2410.07087*, 2024. 1

[37] Jianhao Yuan, Shuyang Sun, Daniel Omeiza, Bo Zhao, Paul Newman, Lars Kunze, and Matthew Gadd. Rag-driver: Generalisable driving explanations with retrieval-augmented in-context multi-modal large language model learning. In *Robotics: Science and Systems*, 2024. 3

[38] Junbo Zhang and Kaisheng Ma. Mg-vln: Benchmarking multi-goal and long-horizon vision-language navigation with language enhanced memory map. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7750–7757. IEEE, 2024. 1

[39] Weichen Zhang, Chen Gao, Shiquan Yu, Ruiying Peng, Baining Zhao, Qian Zhang, Jinqiang Cui, Xinlei Chen, and Yong Li. CityNavAgent: Aerial vision-and-language navigation with hierarchical semantic planning and global memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31292–31309, Vienna, Austria, 2025. Association for Computational Linguistics. 2, 3, 4, 5, 6, 8

[40] Ganlong Zhao, Guanbin Li, Jia Pan, and Yizhou Yu. Aerial vision-and-language navigation with grid-based view selection and map construction. *arXiv preprint arXiv:2503.11091*, 2025. 2, 6

[41] Ganlong Zhao, Guanbin Li, and Yizhou Yu. Navgemini: a multi-modal llm agent for vision-and-language navigation. *Visual Intelligence*, 4(1):1, 2026. 3

[42] Xufeng Zhao, Mengdi Li, Cornelius Weber, Muhammad Burhan Hafez, and Stefan Wermter. Chat with the environment: Interactive multimodal perception using large language models. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3590–3596. IEEE, 2023. 3

[43] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7641–7649, 2024. 6